

AMENDMENTS TO THE CLAIMS

Applicant submits below a complete listing of the current claims, including marked-up claims with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing. This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of the Claims

1. (Previously presented) An assembler for a target microprocessor to automatically track changes in an instruction set of the target microprocessor, the assembler comprising:

a descriptor file containing information descriptive of the instruction set of said target microprocessor, wherein information about a starting position of at least one bit field and a number of bits available for the at least one bit field for at least one instruction in the instruction set is derived and stored as constraints due to the instruction set;

a translation device of the assembler for translating assembly language instructions into machine language as an output;

a fetching device of the assembler for deriving information from the assembly language instructions and acquiring data from said descriptor file using the information, wherein the data is representative of the constraints due to the instruction set and includes selected data comprising at least one encoding function for the at least one instruction due to the changes in the instruction set;

a control device of the assembler arranged to receive said data from said fetching device and said machine language from said translation device, and operable to constrain the machine language to conform to an architecture of said instruction set using the data received from the fetching device; and

a data transfer device of the assembler arranged to output the selected data fetched from said descriptor file directly to a linker to enable the linker to perform operations on the constrained machine language using the selected data, including operations on external symbols using the at least one encoding function.

2. (Previously presented) The assembler of claim 1 wherein the descriptor file comprises syntax information for each instruction, and the control device constrains each instruction on the basis of said syntax information.

3. (Previously presented) A system for assembling a machine language program, comprising the assembler of claim 1 and further comprising a data capture device having an input for accessing the instruction set of said target microprocessor and having an output, wherein said output comprises said descriptor file.

4. (Previously presented) A system for assembling a machine language program, comprising the assembler of claim 1, wherein the linker receives the selected data fetched from the descriptor file by the data transfer device and the machine language constrained by the control device to perform operations on the constrained machine language according to the selected data.

5. (Previously presented) A method of assembling a machine language program for a target microprocessor comprising:

providing a descriptor file containing information descriptive of an instruction set of said target microprocessor, wherein information about a starting position of at least one bit field and a number of bits available for the at least one bit field for at least one instruction in the instruction set is derived and stored as constraints due to the instruction set;

translating assembly language instructions into machine language wherein the translation comprises:

directly transliterating the assembly language instructions to machine language;

deriving information from the assembly language instructions and acquiring data from said descriptor file using the information, wherein the data is representative of the constraints due to the instruction set and includes selected data comprising at least one encoding function for the at least one instruction due to changes in the instruction set;

constraining the directly transliterated machine language to conform to an architecture of said instruction set, thereby assembling the machine language program for the target microprocessor, using the data acquired from the descriptor file; and

transferring the selected data acquired from said descriptor file directly to a linker to enable the linker to perform operations on the constrained machine language using the selected data, including operations on external symbols using the at least one encoding function.

6. (Previously presented) A method of claim 5 wherein said descriptor file contains syntax information for each instruction of the instruction set, and said constraining comprises constraining each assembly language instruction using said syntax information.

7. (Previously presented) A method of preparing a program executable on a target microprocessor comprising:

capturing data from an instruction set of said target microprocessor thereby forming a descriptor file containing information descriptive of said instruction set, wherein information about a starting position of at least one bit field and a number of bits available for the at least one bit field for at least one instruction in the instruction set is derived and stored as constraints due to the instruction set;

providing assembly language instructions for said target microprocessor;

translating each assembly language instruction into a corresponding machine language output;

deriving information from the assembly language instructions and acquiring data from the descriptor file using the information, wherein the data is representative of the constraints due to the instruction set and includes selected data comprising at least one encoding function for the at least one instruction due to changes in the instruction set;

using the data acquired from said descriptor file, constraining the machine language output to conform to an architecture of said instruction set; and

transferring the selected data acquired from said descriptor file directly to a linker to enable the linker to perform operations on the constrained machine language using the selected data, including operations on external symbols using the at least one encoding function.

8. (Previously presented) A method of preparing a program executable on a microprocessor, comprising:

providing plural program modules, at least one of said modules having one or more instructions including external symbols, wherein external symbols have values which cannot be determined without reference to another program module;

providing a descriptor file containing information descriptive of an instruction set of said target microprocessor, wherein information about a starting position of at least one bit field and a number of bits available for the at least one bit field for at least one instruction in the instruction set is derived and stored as constraints due to the instruction set;

translating assembly language instructions into machine language wherein the translation comprises:

directly transliterating the assembly language instructions into machine language;

deriving information from the assembly language instructions and acquiring data from said descriptor file using the information, wherein the data is representative of the constraints due to the instruction set and includes selected data comprising at least one encoding function for the at least one instruction due to changes in the instruction set; and
constraining the directly transliterated machine language to conform to an architecture of said instruction set using the data acquired from the descriptor file;

transferring the selected data acquired from said descriptor file directly to a linker to enable the linker to perform operations on the constrained machine language using the selected data; and

binding external symbols to addresses using the at least one encoding function from the selected data acquired from said descriptor file, thereby preparing the program executable on the microprocessor.

9-11. (Canceled)

12. (Previously presented) The assembler of claim 1, wherein a decoding function is provided, the decoding function is used to check for at least one error in the assembly language instructions.